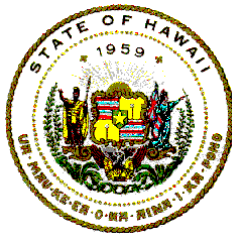


# VSAM STANDARDS

---

December 2003

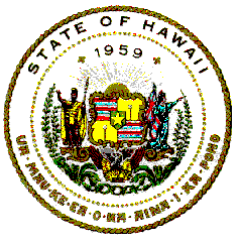


Department of Accounting and General Services  
Information and Communication Services Division

## Information Technology Standards

### TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	PURPOSE .....	1
1.2	SCOPE .....	1
1.3	APPLICABILITY .....	2
1.4	HARDWARE ENVIRONMENT .....	2
1.5	COMMENTS AND SUGGESTIONS .....	2
<b>2</b>	<b>VIRTUAL STORAGE ACCESS METHOD (VSAM) .....</b>	<b>4</b>
2.1	VSAM ORGANIZATION TYPES .....	4
2.2	VSAM ENVIRONMENT .....	4
2.3	CREATING A VSAM DATASET .....	5
2.4	RETRIEVING EXISTING VSAM DATASET .....	5
2.5	CLOSING A VSAM DATASET .....	6
<b>3</b>	<b>VSAM OPTIONS .....</b>	<b>7</b>
3.1	SPEED OR RECOVERY (LOADING OPTION).....	7
3.2	FILE STATUS CLAUSE (COBOL).....	7
3.3	CONTROL INTERVAL SIZE (KSDS DATASETS) .....	8
3.5	IMBED AND REPLICATE (KSDS INDEX).....	9
3.6	UNIQUE OR SUBALLOCATION PARAMETER .....	9
3.7	VERIFY (DATASET RECOVERY) .....	9
<b>4</b>	<b>VSAM FEATURES .....</b>	<b>11</b>
4.1	VSAM SECURITY FACILITIES .....	11
4.2	VSAM BACKUP .....	11
	APPENDIX A -VSAM FOR COBOL .....	14



## **Information Technology Standards**

---

# **1 INTRODUCTION**

The IBM utility, IDCAMS, is a comprehensive and efficient software tool for mainframe computers that use high performance, high capacity direct-access storage devices (DASD).

This IBM access method system (IDCAMS) manages the storage of magnetic disk data access in two formats, the first is "native storage access method," and the second is, "Virtual Storage Access Method" (VSAM).

VSAM is a very flexible tool and with this flexibility comes many optional features. The rules, conventions, and guidelines presented in this document must be followed to achieve the most effective results from the State's mainframe computers.

## **1.1 Purpose**

The purpose for this document is to establish policies, standards, procedures, conventions, and guidelines that will optimize and standardize the State's central site mainframe VSAM systems environments.

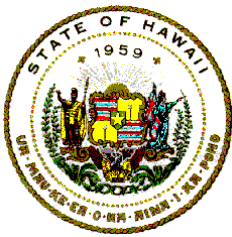
This document has the official State Executive Branch's conventions, rules, and guidelines for the effective use of VSAM at the State's central computer center.

This document will guide computer programmers and data processing systems analysts in the most effective use of VSAM file organization, structure, and resources for application programs and job streams processed at the State of Hawaii's central computer site.

There are some MVS JCL DD (Data Definition) statement options and parameter conventions that are required to support the State's VSAM environment, resources, and facilities. They are presented in this document to give computer programmers and data processing systems analysts a common understanding of what the Department of Accounting and General Services (DAGS) Information and Communication Services Division (ICSD) expects when application programs or job streams that reference VSAM files are submitted for processing at the State's central computer site.

## **1.2 Scope**

The scope of this document is limited to establishing policies, procedures, standards, conventions, and guidelines for consultants, vendors, contractors, computer programmers, and data processing systems analysts to follow when they must set up and use VSAM datasets for application programs or job streams.



## **Information Technology Standards**

---

This document is not a replacement for the vendor's VSAM User and/or reference manuals, or for ICSD sponsored or approved training courses developed for a basic understanding of VSAM.

Non-compliance with these established conventions, rules, or guidelines might result in computer systems error messages or unpredictable results.

### **1.3 Applicability**

The intent of this document is to provide guidance for computer programmers, data processing systems analysts, contractors, and consultants who must develop, test, and install programs, systems, and/or job streams that use VSAM datasets or VSAM space in State controlled computers.

State computer programmers and data processing systems analysts and any contractor or consultant who uses the State's mainframe computing resources at the ICSD central site must follow the standards, procedures, conventions, and guidelines presented in this document.

Failure to follow these procedures, standards, conventions, and/or guidelines may result in delays and/or cancellation of the program or job stream.

### **1.4 Hardware Environment**

The State of Hawaii Executive Branch's central computer center has two general computer systems environments for application system development and testing: the mainframe environment and several server environments. This document will focus on VSAM, as it should be used in the mainframe environment.

### **1.5 Comments and Suggestions**

Any State of Hawaii Information Technology Standards document, reference manual or users guide mentioned in this document are available through the departmental user agency data processing coordinator (DP Coordinator). Standards are also accessible on-line by clicking on [Information Technology Standards](#) on the [ICSD](#) home page at:

<http://www.hawaii.gov/icsd/>

Statewide Forms are accessible on-line by clicking on [Forms Central](#) on the [Government in Hawaii](#) home page at:

<http://www.ehawaii.gov/government/html/>



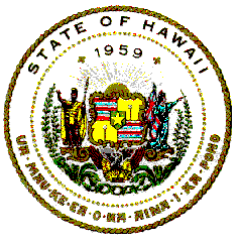
Department of Accounting and General Services  
Information and Communication Services Division

## **Information Technology Standards**

---

Comments, recommendations, proposals, or suggestions regarding the contents of this document may be sent either via email to [icsd.admin.ppmo@hawaii.gov](mailto:icsd.admin.ppmo@hawaii.gov) or in writing to:

Information and Communication Services Division  
Project Planning and Management Office  
1151 Punchbowl Street, B10  
Honolulu, Hawaii 96813-3024



## **2 VIRTUAL STORAGE ACCESS METHOD (VSAM)**

VSAM is an IBM/MVS Operating System's access method system; it is not a data base management system. VSAM supports batch users, on-line transactions, and data base applications.

VSAM is different from all other IBM access methods. Certain options and precautions must be evaluated and considered when VSAM datasets are defined and created. There are almost thirty different parameters that may be specified for data access and control options for the VSAM file.

### **2.1 VSAM Organization Types**

VSAM supports the following three (3) file organization types.

#### **2.1.1 ESDS**

The "entry-sequence dataset" (ESDS) is like a standard ("flat-file") sequentially accessed (SAM or QSAM) dataset.

#### **2.1.2 RRDS**

The "relative-record dataset" (RRDS) is like a directly accessible dataset file (DAM or RDAM).

#### **2.1.3 KSDS**

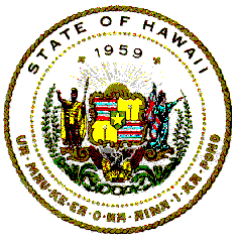
The "key-sequenced dataset" (KSDS) can be processed sequentially or randomly based on the value of a key within a data record.

Records in a KSDS may also be accessed via an "alternate index" which specifies an order different from the file's primary (base) record key.

Although an alternate index may have an ESDS base cluster, most alternate indexes are built over a KSDS cluster.

### **2.2 VSAM Environment**

In VSAM terms, a file is called a cluster. A cluster is a set of catalog entries that represent a file. A cluster has a data component that has the actual records of the file, and it may have an index component for the indexes for a KSDS file.



Department of Accounting and General Services  
Information and Communication Services Division

## Information Technology Standards

---

VSAM space is an area of a disk volume that's under the control of VSAM, so to the operating system, VSAM space is just another file. Within a VSAM space, VSAM can create sub-allocated files. If an entire space is to be allocated to a single VSAM file, it is called a "UNIQUE" file.

### 2.3 Creating a VSAM Dataset

The MVS JCL DD statement cannot be used to create a new VSAM dataset. The IBM Access Method Services (IDCAMS) commands must be used to create the space for a VSAM dataset.

Consult the IBM "DFMS MVS Access Method Services" reference manual for information to create VSAM datasets.

The following is an example of the "DEFINE CLUSTER" command to create either a hypothetical KSDS or ESDS file.

```
DEFINE      CLUSTER  ( NAME(PMS.PROJECT.MASTER)      -  
                      INDEXED      -  
                      RECORDSIZE(200 200) -  
                      KEYS(6 0) -  
                      VOLUMES(PMSVOLS) -  
                      UNIQUE )  
              DATA   ( NAME(PMS.PROJECT.MASTER.DATA) -  
                      CYLINDERS(50 5) )  
              INDEX   ( NAME(PMS.PROJECT.MASTER.INDEX) )  
  
DEFINE      CLUSTER  ( NAME(PMS.PROJECT.TRANS)-  
                      NONINDEXED      -  
                      RECORDSIZE(190 280) -  
                      VOLUMES(PMSVOLS) )  
              DATA   ( NAME(PMS.PROJECT.TRANS.DATA)-  
                      CYLINDERS(10 1) )
```

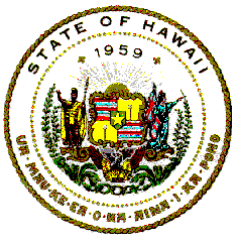
Do not include the CATALOG parameter in the DEFINE command.

### 2.4 Retrieving Existing VSAM Dataset

To request access to a previously created VSAM dataset, an MVS JCL DD statement using one of the following formats must be in the JCL job stream:

```
//ddname    DD    DSNAME=datasetname,
```





Department of Accounting and General Services  
Information and Communication Services Division

## Information Technology Standards

---

```
//          DISP=SHR,  
//ddname    DD    DSNAME=datasetname,  
//          DISP=OLD,
```

A VSAM dataset cannot be passed to another step within a job stream.

The option "DISP=SHR" means that the dataset could be shared by other jobs if the share option was specified when the file was defined.

The option "DISP=OLD" means that the dataset would not be shared by other jobs even if the share option had been specified when the file was defined.

### 2.5 Closing a VSAM Dataset

Under normal termination situations, the CLOSE macro does the following:

- a. Disconnects the application program from the dataset.
- b. Causes VSAM to update information in the catalog.
- c. Writes out buffers of data or index records when their contents were changed by the job but which were not already written out.

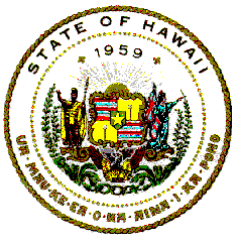
If an application program had abnormally terminated, all datasets that were open at the time of the ABEND would be closed.

The VSAM CLOSE invoked by an ABEND will not update the dataset's catalog information. It does not complete outstanding I/O requests, and buffers are not flushed. This means that the catalog might not properly reflect the cluster's status, and the index might not accurately reference some of the data records.

If a VSAM dataset is closed as a result of an ABEND, the programmer should execute an IDCAMS VERIFY command to restore the dataset's end-of-file values before the dataset is accessed again.

The VERIFY command does not work for KSDS or RRDS files that were improperly closed when they had been open for a load operation. If a load does ABEND, the programmer must delete the cluster, redefine it, and start the load operation over from the beginning.





### **3 VSAM OPTIONS**

Many options, features, and situations either influence or, in some cases, determine VSAM's performance as well as the performance of the computer's operating system.

#### **3.1 Speed or Recovery (Loading Option)**

The SPEED or RECOVERY option is coded at the CLUSTER or DATA level of a DEFINE CLUSTER command. SPEED or RECOVERY applies only to the initial loading of data. This option influences how VSAM loads data into a new file.

- a. If RECOVERY is specified (or allowed as default), VSAM pre-formats each control area that it allocates with binary zeros before it writes any control intervals to it.
- b. If SPEED is specified, VSAM does not pre-format the control areas. Therefore, the data load processing is much faster when SPEED is in effect because the additional I/O required to pre-format the control intervals is eliminated.

Since dataset load is probably not a significant part of the total job processing time, the SPEED option should be used instead of the RECOVERY option.

The overhead involved with the RECOVERY option is normally not worth the potential savings. If a system failure does occur, the dataset can be deleted, redefined, and loaded again from the beginning.

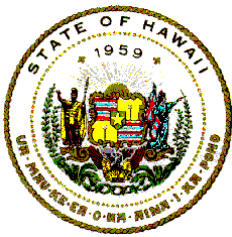
#### **3.2 File Status Clause (COBOL)**

The FILE STATUS clause in a COBOL program specifies a field that is updated by VSAM when an I/O operation for that VSAM file has been executed.

The FILE STATUS field must be defined in the program's Working-Storage Section as a two-character alphanumeric item.

When a COBOL I/O operation is successful, VSAM will put "00" in the FILE STATUS field. But when an error condition occurs, VSAM will put a non-zero two-digit error code in the FILE STATUS field.

Therefore, COBOL program's logic should examine the FILE STATUS field after each VSAM I/O operation to determine whether an error occurred and, if so, what action should be taken (See the Appendix for more detailed information).



The available COBOL VSAM I/O operation verbs are:

OPEN, CLOSE, READ, WRITE, DELETE, REWRITE, and START.

### **3.3 Control Interval Size (KSDS Datasets)**

One of the most important aspects of insuring good VSAM file performance is selecting an appropriate control interval (CI) size. To specify a control interval size, code the CONTROL INTERVAL SIZE parameter (abbreviated CISZ) in the DEFINE command.

Although the CISZ parameter can be coded at the CLUSTER level, that's not a good idea because VSAM would assign the same CI size to both the data component and index component.

It is recommended that the CISZ be specified at the DATA level, and that VSAM should calculate the index CI size.

- a. Initially for most datasets, CISZ(4096) is an appropriate assignment. Smaller CI size (4096 or less) is recommended when a file will be mostly used for random retrieval. Larger CI size is recommended when the file will be processed sequentially. CI size up to 8192 bytes must be a multiple of 512. CI size larger than 8192 must be a multiple of 2048.
- b. When VSAM tries to insert a record into a control interval that does not have enough free space, VSAM looks for a free control interval in the same control area so that it can do a CI split. If there are no free control intervals in that control area, VSAM splits the entire control area and may even extend the dataset.

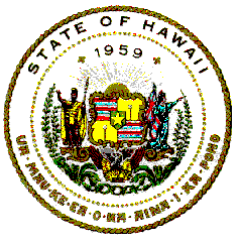
If the control interval size (CI size) parameter is not explicitly specified in the "DEFINE" COMMAND, VSAM will select the CI size for both the data component and index component.

However, VSAM usually makes a poor choice for the data component CI size, but for the index component, VSAM's CI size value is usually best.

### **3.4 Free space**

Free space is allocated in terms of bytes, and not in terms of logical records. The parameters for defining free space are percentages of the defined data component.

For the FREESPACE parameter, the first value defines the percentage of free space to be reserved within each control interval, and the second defines the number of control



## Information Technology Standards

---

intervals within each control area that should be reserved as free space.

### 3.5 IMBED and REPLICATE (KSDS Index)

The DEFINE CLUSTER parameters IMBED and REPLICATE affects the way that the index component of a keyed sequence dataset (KSDS) is stored. The index component has two parts, an index set record, and a sequence set record.

The IMBED and REPLICATE options may be coded in either the CLUSTER or the INDEX level.

- a. The IMBED option places the index component's sequence set records in the space with the data component and causes the sequence set record to be duplicated as many times as possible on its track. This means that the first track of each control area will not be available for user data.
- b. The REPLICATE option places both index set and sequence set records on a separate track, duplicated as many times as possible.

The REPLICATE option may be used with the IMBED option; however, the IMBED option without the REPLICATE option, and with sufficient buffering for the index is effective for most situations.

The easiest way to specify (and adjust) buffer space for the data and index is to specify BUFND and BUFNI in the AMP parameter of the file's JCL DD statement.

### 3.6 UNIQUE or SUBALLOCATION Parameter

The UNIQUE or SUBALLOCATION parameter specifies whether a file should be sub-allocated within existing VSAM space, or created in its own space.

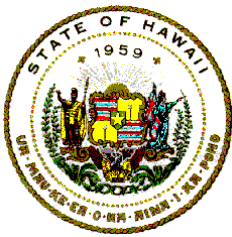
The default attribute is SUBALLOCATION.

At ICSD, the UNIQUE parameter must be specified when files are defined.

### 3.7 VERIFY (Dataset Recovery)

The VERIFY command is used to verify, and if necessary, to update, the end-of-file information in the catalog in order to make the catalog information consistent with the data file.

The VERIFY command cannot be used for an empty VSAM file where the high-used



## Information Technology Standards

---

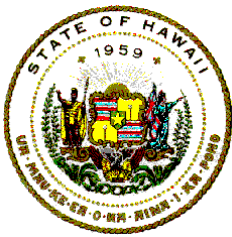
Relative Byte Address (RBA) in its catalog record is zero (0).

When a VSAM dataset is closed in an update program, VSAM will update the end-of-file information in both the catalog and the data file.

If an update program fails or the operating system fails, VSAM may not close the file properly. When this happens, VSAM is unable to update important information in the file's catalog record.

The VSAM file's catalog record has the high-used RBA that specifies the end of the file address. If this field is not updated, the information stored in the catalog record does not agree with the actual contents of the related file.

After the abnormal termination of a VSAM file, the VERIFY command must be executed to correct the catalog record information to reflect the actual status of the dataset before the file is used again.



## **4 VSAM FEATURES**

Certain features, facilities, functions, structures, or environments affect the efficiency and effectiveness of VSAM's operations and the performance of the computer's operating system.

### **4.1 VSAM Security Facilities**

The security facilities provided by VSAM generally are not sufficient for an installation that is serious about data security. For one thing, too many passwords are involved. With VSAM, it is possible to assign as many as twelve (12) different passwords to each VSAM KSDS (four each for the cluster, the data component, and the index component). As a result, a shop can easily end up managing thousands of different passwords.

Because of the inherent weaknesses of VSAM security, most installations use other security managers that provide a more comprehensive, system-wide approach to security.

At ICSD on the MVS systems, the IBM product Resource Access Control Facility (RACF) is used for data security.

### **4.2 VSAM Backup**

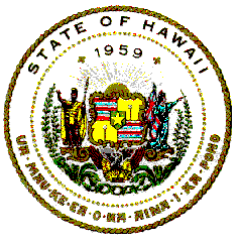
Most installations routinely back up entire DASD volumes. Strictly speaking, volume-level backup and restoration is not a VSAM consideration; AMS does not perform it.

ICSD has a utility program, FDR, which can dump disk volumes to magnetic tapes and can restore disk volumes from backup magnetic tapes. It can be used to back up disk volumes that contain VSAM data.

Application project managers must arrange to have their critical VSAM files backed up routinely. There are three (3) ways to back up VSAM files:

- a. A batch processing application designed to backup the data.
- b. The EXPORT and IMPORT commands to produce a backup copy of a dataset and later restore that copy.
- c. The REPRO command to copy a VSAM dataset along with DEFINE (and possibly DELETE) commands to backup and restores the file.

The recommended way is to use the AMS command REPRO to back up and restore



Department of Accounting and General Services  
Information and Communication Services Division

## Information Technology Standards

---

VSAM files.

Programmers should REPRO to magnetic tape cartridge(s) for their backups, and use the DEFINE and the DELETE command to restore the data files from the magnetic tape cartridge(s).

When restoring the output file specified in a REPRO command must exist. The output file must have been created with a DEFINE CLUSTER command.

When the output file is empty, VSAM processes the file in load mode, and records are copied one by one from the input file to the output file.

When the output file is not empty, VSAM merges the input file with the output file.

### 4.2.1 BACKUP TO TLMS MEDIA

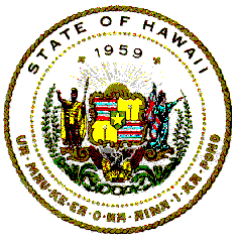
The following is an example that uses the REPRO command to backup a hypothetical file to a TLMS controlled GDG on tape cartridges:

```
//JOB2      JOB  (1234,PMS,2,5),'PROGRAMMER-NAME',
//          MSGLEVEL=(1,1),
//          MSGCLASS=E,
//          CLASS=J
//STEPB     EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//DATAIN    DD  DSN=PMS.MASTER.DATA,
//          DISP=OLD
//DATAOUT   DD  DSN=TMS.PMS.MSTRDATA(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=(ACS,,DEFER),
//          DCB=(X.SHRMOD,RECFM=FB,LRECL=080,BLKSIZE=32720)
//SYSIN     DD  *
            REPRO      INFILE      (DATAIN) -
                        OUTFILE     DATAOUT)

/*
//
```

### 4.2.2 REORGANIZE KSDS

The following is the recommended method that uses the REPRO command to reorganize a hypothetical KSDS file:



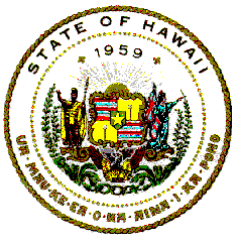
Department of Accounting and General Services  
Information and Communication Services Division

## Information Technology Standards

---

1. REPRO VSAM file to tape.
2. Check for successful execution of REPRO.
3. If successful, DELETE and DEFINE VSAM file.
4. REPRO from tape to VSAM file.





## APPENDIX A -VSAM for COBOL

### Input/Output Statements Common Processing Facilities

The common processing facilities provided are:

- I. status key
- II. invalid key condition
- III. INTO/FROM identifier option
- IV. current record pointer

These facilities are discussed in the following three sections.

#### I. STATUS KEY

If the FILE STATUS clause is specified in the FILE-CONTROL entry, a value is placed in the specified STATUS KEY (the two (2) character data item that is named in the FILE STATUS clause) during the execution of any request on that file.

The value indicates the status of that request.

The value is placed in the STATUS KEY before the execution of any EXCEPTION/ERROR Declaratives, or the INVALID KEY/AT END option associated with the request.

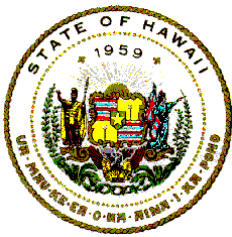
The first character of the STATUS KEY is known as the HIGH-ORDER DIGIT.

The second character is known as the LOW-ORDER DIGIT.

For VSAM files, the combination of possible STATUS KEY values, and their meanings are shown in the next table.

**Status Key Values and Meanings**

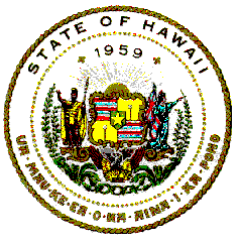
High-Order Digit	MEANING	Low-Order Digit	MEANING
0	Successful Completion	0	No further information
		2	This file status value only applies to indexed files with alternate keys that allow duplicates.  The input-output Statement was successfully executed, but a duplicate key was detected. For a READ statement



Department of Accounting and General Services  
Information and Communication Services Division

## Information Technology Standards

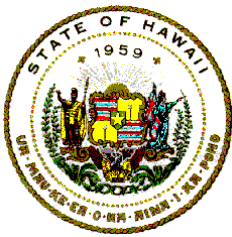
			the key value for the current key of reference was equal to the value of the same key in the next record within the current key of For a REWRITE or WRITE statement, the record just written created a duplicate key value for at least one alternate record key for which duplicates are allowed.
		4	A READ statement was successfully executed, but the length of the record being processed did not conform to the fixed file attributes for that file.
		5	An OPEN statement is successfully executed, but the referenced optional file is not present at the time the OPEN statement is executed. The file has been created if the open mode is I-o or EXTEND. This does not apply to MVS and VM VSAM sequential files.
		7	For a CLOSE statement with NO REWIND, REEL/UNIT, or FOR REMOVAL phrase or for an OPEN statement with the NO REWIND phrase, the referenced file was not on a non-reel/unit medium.
1	At End Condition	0	A sequential READ statement was attempted and no next logical record existed in the file because the end of the file had been reached, or the first READ was attempted on an optional input file that was not present.
		4	Sequential READ statement was attempted for a relative file and the number of significant digits in the relative record number was larger than the size of relative key data item described for the file.
2	Invalid Key Condition	1	A sequence error exists for a sequentially accessed indexed file. The prime record key value has been changed by the program between the successful execution of a READ statement and the execution of the next REWRITE statement for that file, or the ascending requirements for successive record key values were violated.  Under OS/2 for Btrieve indexed files, file status 21 is not applicable. You can create records sequentially in any key order.
		2	An attempt was made to write a record that would create a duplicate key in a relative file; or an attempt was made to write or rewrite a record that would create a duplicate prime record key or a duplicate alternate record key without the DUPLICATES phrase in an indexed file.
		3	An attempt was made to randomly access a record that does not exist in the file, or a START or random READ statement was attempted on an optional input file that was not present.
		4	An attempt was made to write beyond the externally defined boundaries of a relative or indexed file, or a sequential WRITE statement was attempted for a relative



Department of Accounting and General Services  
Information and Communication Services Division

## Information Technology Standards

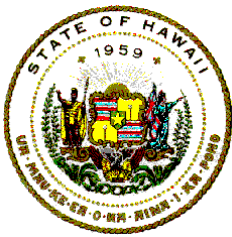
			file and the number of significant digits in the relative record number was larger than the size of the relative key data item described for the file.
3	Permanent Error Condition	0	No further information
		4	A permanent error exists because of a boundary violation; an attempt was made to write beyond the externally defined boundaries of a sequential file.
		5	An OPEN statement with the INPUT, I-O, or EXTEND phrase was attempted on a non-optional file that was not present.
		7	An OPEN statement was attempted on a file that would not support the open mode specified in the OPEN statement.  Possible violations are: 1. The EXTEND or OUTPUT phrase was specified but the file would not support write operations. 2. The I-O phrase was specified but the file would not support the input and output operations permitted. 3. The INPUT phrase was specified but the file would not support read operations.
		8	An OPEN statement was attempted on a file previously closed with lock.
		9	The OPEN statement was unsuccessful because a conflict was detected between the fixed file attributes and the attributes specified for that file in the program. These attributes include the organization of the file (sequential, relative, or indexed), the prime record key, the alternate record keys, the code set, the maximum record size, the record type (fixed or variable), and the blocking factor.  Under AIX, OS/2, and Windows, file status 39 is not supported for line sequential files or Btrieve files.
4	Logic error conditions	1	An OPEN statement was attempted for a file in the open mode.
		2	A CLOSE statement was attempted for a file not in the open mode.
		3	For a mass storage file in the sequential access mode, the last input-output statement executed for the associated file prior to the execution of a REWRITE statement was not a successfully executed READ statement.  For relative and indexed files in the sequential access mode, the last input-output statement executed for the file prior to the execution of a DELETE or REWRITE statement was not a successfully executed READ statement.
		4	A boundary violation exists because an attempt was made



Department of Accounting and General Services  
Information and Communication Services Division

# Information Technology Standards

			to rewrite a record to a file and the record was not the same size as the record being replaced, or an attempt was made to write or rewrite a record that was larger than the largest or smaller than the smallest record allowed by the RECORD IS VARYING clause of the associated file-name.
		6	A sequential READ statement was attempted on a file open in the input or I-O mode and no valid next record had been established because:  1. The preceding READ statement was unsuccessful but did not cause an at end condition. 2. The preceding READ statement caused an at end condition.
		7	The execution of a READ statement was attempted on a file not open in the input or I-O mode.
		8	The execution of a WRITE statement was attempted on a file not open in the I-O, output, or extend mode.
		9	The execution of a DELETE or REWRITE statement was attempted on a file not open in the I-O mode.
9	Implementor-Defined Condition	0	No further information.
		1	For VSAM only on MVS and VM: Password failure.  Under AIX, OS/2, and Windows: Authorization failure.
		2	Logic error.
		3	For all files, except QSAM, resource not available.
		4	For VSAM under MVS and VM with CMPR2 compiler-option only: No file position indicator for sequential request.  Under AIX, OS/2, and Windows: Concurrent open error.
		5	For all files, except QSAM, invalid or incomplete information.
		6	For VSAM file under MVS and VM: An OPEN statement with the OUTPUT or EXTEND phrase was attempted for an optional file, but no DD statement was specified for the file. For QSAM file under MVS and VM: An OPEN statement with the OUTPUT or EXTEND phrase was attempted for an optional file, but no DD statement was specified for the file and the CBLQDA(OFF) run-time option was specified.  Under AIX, OS/2, and Windows: File system not available.
		7	For VSAM only under MVS and VM:



## Information Technology Standards

			OPEN statement execution successful: File integrity verified.
			Under AIX, OS/2, and Windows: Errors related to remote file access.
		8	Under AIX, OS/2, and Windows: Open failed due to locked file.
		9	Under AIX, OS/2, and Windows: Record Access failed due to locked record.

### II. Invalid Key Condition

The INVALID KEY condition can occur during the execution of a START, READ, WRITE, REWRITE, or DELETE statement.

For details of the causes for the condition, see the discussions of those statements.

When the INVALID KEY condition is recognized, the following actions are taken in the following order:

1. If the FILE-STATUS clause is specified in the FILE-CONTROL entry, a value is placed into the STATUS KEY to indicate an INVALID KEY condition.
2. If the INVALID KEY option is specified in the statement causing the condition, control is transferred to the INVALID KEY imperative-statement.

Any EXCEPTION/ERROR declarative procedure specified for this file is not executed.

3. If the INVALID KEY option is not specified, but the EXCEPTION/ERROR declarative procedure is specified for the file, the EXCEPTION/ERROR procedure is executed.

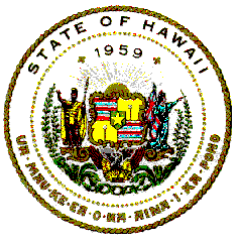
When an INVALID KEY condition occurs, the Input/Output statement that caused the condition is not successful.

If the INVALID KEY option is not specified for the file, an EXCEPTION/ERROR procedure must be specified.

(HOWEVER (as an IBM Extension), this implementation allows both the INVALID KEY option, and the EXCEPTION/ERROR procedure to be omitted.)

### III. INTO/FROM IDENTIFIER OPTION

This option is valid for READ, REWRITE, and WRITE statements.



## Information Technology Standards

---

The specified identifier must be the name of an Entry in the WORKING-STORAGE SECTION, the LINKAGE SECTION, or of a record description for another previously opened file.

RECORD-NAME/FILE-NAME and the IDENTIFIER must not refer to the same Storage Area.

The statements take the following form:

READ file-name RECORD INTO identifier

WRITE record-name FROM identifier

REWRITE record-name FROM identifier